ยืน ภู่วรวรรณ

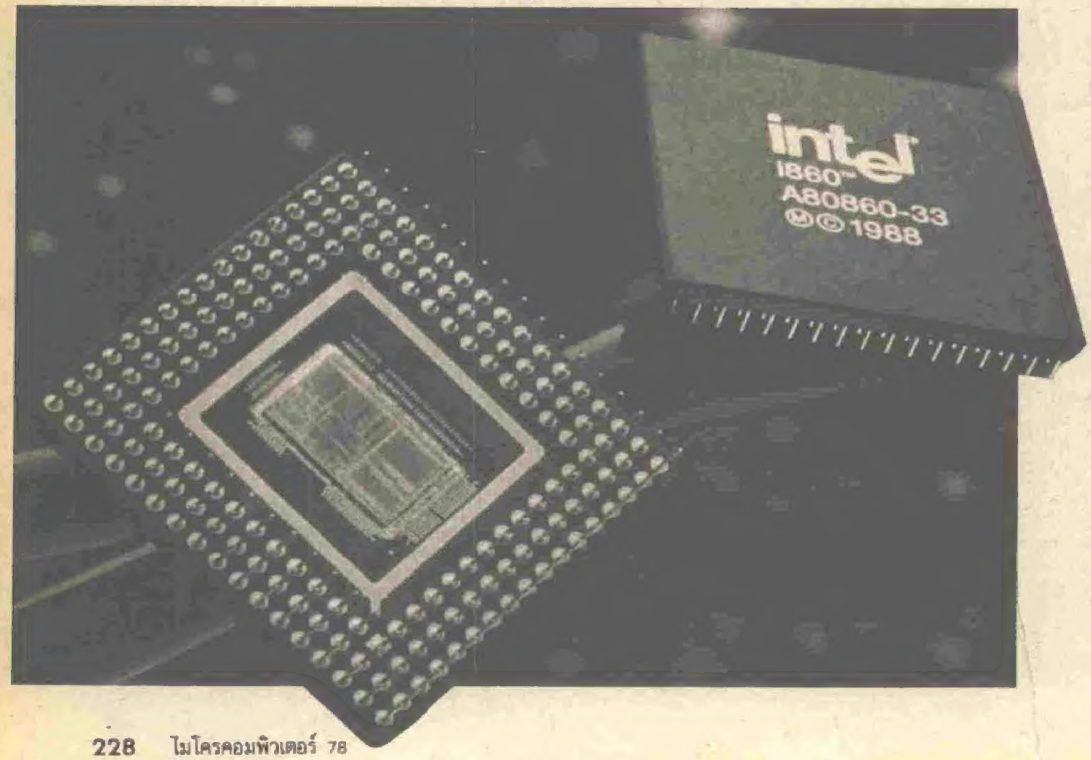
RISC
พัฒนาการที่ไม่หยุดยั้ง

ความเป็นมาของสถาปัตยกรรมแบบ RISC และแนวโน้มของการนำมาใช้ในอนาคต

ผู้ออกแบบคอมพิวเตอร์มีทางเลือกหลายทางที่จะออกแบบให้คอมพิวเตอร์มีประสิทธิภาพดีและทำงานได้เร็วที่สุด การเลือกโครงสร้างสถาปัตยกรรมในรูปแบบใดขึ้นกับความต้องการและสภาพของการประยุกต์ใช้งาน ในช่วงหลังจากปี ค.ศ. 1980 ผู้ออกแบบคอมพิวเตอร์ได้เพ่งเล็งมาที่โครงสร้างสถาปัตยกรรมแบบ RISC มาก ผู้ออกแบบหลายค่ายพบว่า การใช้สถาปัตยกรรมแบบ RISC ทำให้ซีพียูทำงานได้เร็วขึ้น การออกแบบง่ายขึ้นและต้นทุนการผลิตถูกกว่าแบบเดิม

RISC เป็นคำย่อมาจาก Reduce Instruction Set Computer ซึ่งก็คือคอมพิวเตอร์ที่ผู้ออกแบบพยายามลดชุดคำสั่งลงให้เหลือเท่าที่จำเป็นและลดให้น้อยลง โดยมีสถาปัตยกรรมสวนทางกับแบบเก่า คือพยายามเพิ่มชุดคำสั่งให้มากขึ้น เพื่อตอบสนองการทำงานเฉพาะบางอย่าง จึงเรียกคอมพิวเตอร์แบบนั้นว่า CISC (Complex Instruction Set Computer)

การออกแบบ RISC จึงเป็นเรื่องที่เกิดจากแนวความคิดที่จะทำให้คอมพิวเตอร์ทำงานได้เร็วขึ้น โดยเฉพาะอย่างยิ่งงานการคำนวณทางด้านคณิตศาสตร์ และความคิดนี้ดูจะเป็นจริงจังมากขึ้นในยุคของเครื่องคอมพิวเตอร์แบบเวอร์กสเตชันที่กำลังมาแรง และโครงสร้างของการออกแบบชิพทำได้ดีขึ้นจนสามารถออกแบบซีพียูที่สลับซับซ้อนขึ้นได้



จุดเริ่มต้นและความเป็นมา

ในปี ค.ศ. 1948 ครั้งที่เริ่มมีการพัฒนาดิจิตอลคอมพิวเตอร์ การพัฒนาในครั้งนั้นเริ่มจากเครื่อง MARK 1 ซึ่งมีคำสั่งเพียง 7 คำสั่ง แต่ละคำสั่งมีลักษณะการทำงานที่ซับซ้อน เช่น คำสั่งบวก คำสั่งกระโดด (jump) คำสั่งในยุคนั้นมีเพียงเท่าที่จำเป็น แต่เมื่อมีการพัฒนากันต่อมาเรื่อย การสร้างคำสั่งก็เพิ่มมากขึ้น เช่น VAX มีคำสั่งถึงกว่า 300 คำสั่ง นอกจากนี้คำสั่งก็ดูจะซับซ้อนมากขึ้น เช่น คำสั่งแทรกข้อมูลลงในลิสต์ของกลุ่มข้อมูล หรือคำสั่งการคำนวณโพลีโนเมียลของตัวเลขจำนวนจริง

หากพิจารณาไมโครโปรเซสเซอร์ 8080 มีเพียง 78 คำสั่ง แต่เมื่อ Z80 ได้รับการพัฒนาขึ้น ชุดคำสั่งก็เพิ่มเป็นซูเปอร์เซตคือมีถึงกว่า 150 คำสั่ง พัฒนาการต่อมาเป็น 8088, 8086 ก็มีชุดคำสั่งเพิ่มขึ้นอีก การออกแบบแต่ละครั้งดูจะพยายามสร้างชิพเดิมให้เป็นชิพย่อยและทำงานได้เสมือนเป็นสับเซต ดังนั้นคำสั่งที่พัฒนากันต่อมาก็ยิ่งสลับซับซ้อนยิ่งขึ้น หรือที่เรียกว่า CISC นั่นเอง

อย่างไรก็ตาม การพัฒนา CISC ก็ได้ใช้เทคนิคการเพิ่มประสิทธิภาพ โดยรวมหลายอย่างที่ทำให้การพัฒนาในยุคหลังคือ RISC นำมาใช้ เช่นมีการทำ pipelining การพรีเฟทช์ และการใช้แคช

ทำไมผู้ออกแบบคอมพิวเตอร์จึงเลือกเดินทางแบบ CISC

การที่พัฒนาการของคอมพิวเตอร์เป็นไปตามเส้นทาง CISC คือ ทำให้มีชุดคำสั่งที่สับซับซ้อนมากยิ่งขึ้น และ

เหตุที่ทำให้ผู้พัฒนาเลือกเดินทางในแนวนี้เพราะ

ความต้องการให้คอมแพตทิเบิลกับของเดิม

เมื่อฐานของผู้ใช้มีมาก ผู้ใช้คุ้นเคยกับซีพียูหรือคำสั่งของซีพียูนั้นมากแล้ว ครั้นพัฒนาการต้องการทำให้ซีพียูดีขึ้น ก็ต้องทำให้คอมแพตทิเบิลของเดิม และเพิ่มคำสั่งต่าง ๆ เข้าไปอีก การเพิ่มเติมขีดความสามารถบางสิ่งบางอย่างเข้าไปอีกนี้ทำให้เกิดความซับซ้อนในตัวซีพียูมากขึ้น คำสั่งที่เพิ่มเติมทีหลังจึงเป็นคำสั่งพิเศษ โดยที่ไม่ไปกระทบกับซอฟต์แวร์เดิมที่มีอยู่แล้ว

ผู้ออกแบบต้องการลดช่องว่างของซอฟต์แวร์ ระหว่างภาษาชั้นสูงกับภาษาเครื่อง

ด้วยความพยายามที่จะสร้างคำสั่งภาษาเครื่องให้ใกล้หรือเข้าหาภาษาระดับสูงให้มากที่สุด โปรแกรมเมอร์ที่เขียนภาษาสูง เมื่อคอมพายเลอร์แปลความมาเป็นภาษาเครื่องก็แปลได้โดยตรงหรือง่ายขึ้น ดังนั้นคำสั่งในภาษาเครื่องจึงต้องสร้างให้ซับซ้อนเพื่อรองรับภาษาระดับสูง

การออกแบบโดยใช้ไมโครโค้ด

เพื่อให้ฟังก์ชันในระดับสูงของซอฟต์แวร์มาเป็นไมโครโค้ด หรือจากไมโครโค้ดมาเป็นฮาร์ดแวร์ได้โดยตรง เช่น การมีโปรเซสเซอร์คณิตศาสตร์ที่ทำฟังก์ชันเทียบเท่ากับโปรแกรมย่อยของภาษาระดับสูง เช่น การหาค่า log, sin, cos เป็นต้น การสร้างไมโครโค้ดเข้าสู่ฮาร์ดแวร์จึงทำให้กลไกการทำงานซับซ้อนมากขึ้น

การมีเครื่องมือ CAD ช่วยออกแบบคอมพิวเตอร์

จึงทำให้ผู้ออกแบบออกแบบคอมพิวเตอร์ได้ซับซ้อนยิ่งขึ้น เป็นไปในลักษณะตามวัตถุประสงค์เฉพาะได้ CAD จึงมีผลทำให้การออกแบบชิพหรือโครงสร้างสถาปัตยกรรมที่ซับซ้อนได้

จากแนวความคิดที่พยายามทำให้คอมพิวเตอร์ทำงานเร็วขึ้น โดยพยายามคิดหาสถาปัตยกรรมคอมพิวเตอร์ที่เหมาะสม ในราวปี ค.ศ. 1972 จอร์ช คอค (Cocke) แห่งไอบีเอ็ม ได้เฝ้าสังเกตการทำงานของเครื่องคอมพิวเตอร์ไอบีเอ็ม 360 พบว่ามีคำสั่งที่ยุ่งยากและต้องใช้เวลาทำงานแต่ละคำสั่งหลายลูกของสัญญาณนาฬิกา แต่ละคำสั่งก็สั้นยาวต่างกัน เขามีความคิดว่าเป็นไปได้หรือไม่ที่จะทำให้คำสั่งของคอมพิวเตอร์ไม่ต้องวิ่งในระดับไมโครโค้ดที่ต้องใช้สัญญาณนาฬิกาหลายลูก แต่ให้ทำเสร็จในสัญญาณนาฬิกาลูกเดียว ผลของความคิดนี้จึงเกิดการรวมกลุ่มวิจัยระบบ 801 ขึ้น โดยอาศัยพื้นฐานของแนวความคิดดังนี้

หนึ่งคำสั่งใช้เวลาหนึ่งลูกของสัญญาณนาฬิกา

ทีมงานต้องการหาโครงสร้างของฮาร์ดแวร์ที่ทำให้การทำงานของคอม-

พิวเตอร์ทำงานแต่ละคำสั่งได้เสร็จในเวลาหนึ่งลูกของสัญญาณนาฬิกา นั่นคือตัว 10 MHz ก็ทำได้ 10 ล้านคำสั่งในเวลา 1 วินาที ด้วยความคิดนี้ก็ต้องทำคำสั่งให้ง่ายขึ้นไม่ยุ่งยากซับซ้อน

โครงสร้างหน่วยความจำแบบลำดับ

เพื่อให้ซีพียูไม่เสียเวลาในการอ่านข้อมูลจากหน่วยความจำ หรือต้องรอเวลา ดังนั้นโครงสร้างหน่วยความจำต้องมีลำดับคือสามารถเข้าถึงได้ทันทีและรวดเร็ว แนวความคิดเรื่องแคชจึงเกิดขึ้น

คอมพายเลอร์เป็นสิ่งที่นักโปรแกรมต้องพึ่ง

เมื่อชุดคำสั่งมีเพียงพื้นฐานการเขียนโปรแกรมจึงต้องอาศัยคอมพายเลอร์ที่ดี ผู้เขียนโปรแกรมจึงเขียนโปรแกรมในระดับสูง แล้วให้คอมพายเลอร์แปลงไปเป็นคำสั่งในระดับภาษาเครื่องให้

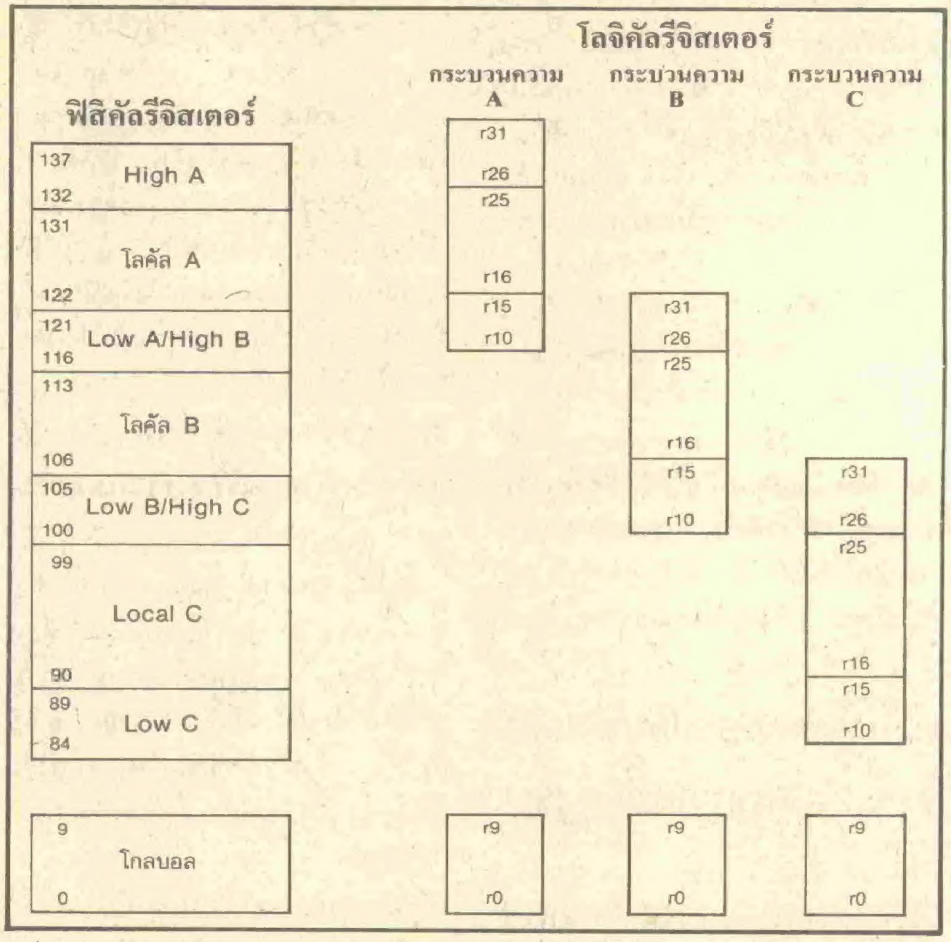
เริ่มต่อในมหาวิทยาลัย

ในราวปี ค.ศ. 1975 มหาวิทยาลัยแคลิฟอร์เนีย ที่เบอร์กเล่ย์ ก็สร้างโครงงานพัฒนาคอมพิวเตอร์โดยใช้ชื่อโครงงานว่า RISC โครงงานนี้สร้างเพื่อให้นักเรียนที่เรียนในระดับบัณฑิตศึกษาได้ทำวิทยานิพนธ์เกี่ยวกับโครงสร้างสถาปัตยกรรมของคอมพิวเตอร์เหล่านี้เพื่อหาประสิทธิภาพที่เพิ่มขึ้น ต่อมาได้เพิ่มโครงงาน SOAR และใช้ CAD ช่วยในการออกแบบและสร้างชิพ

RISC I เป็นซีพียูที่มีชุดคำสั่งที่ทำงานด้วยเวลาสั้น ๆ ดังนั้นการประมวลผลข้อมูลจะต้องทำกับรีจิสเตอร์ จะมีคำสั่งที่เข้าถึงหน่วยความจำเพียงการโหลดและสตอร์เท่านั้น การทำงานของคำสั่งส่วนใหญ่เสร็จในหนึ่งไซเกิล คำสั่งที่ใช้มี 31 คำสั่ง ซึ่งสามารถบรรจุเป็นรหัสขนาดเพียง 32 บิต การถอดรหัสคำสั่งทำได้ทันทีด้วยฟอร์แมตที่แน่นอน ลักษณะพิเศษของ RISC I คือมีรีจิสเตอร์ภายในมาก โดยมีมากกว่าร้อยรีจิสเตอร์ โดยรีจิสเตอร์เหล่านั้นจัดเรียงเป็นกลุ่มและทับซ้อนกัน เพื่อให้มีการเรียกข้อมูลมาใช้หรือส่งผ่านข้อมูลถึงกันได้โดยไม่ต้องเสียเวลาไปรับจากหน่วยความจำ RISC I จึงเป็นซีพียูที่ลดการติดต่อระหว่างซีพียูกับหน่วยความจำ

RISC I มีรีจิสเตอร์ขนาด 32 บิต ทั้งสิ้น 138 รีจิสเตอร์ แบ่งรีจิสเตอร์ออกเป็นกลุ่ม ๆ เรียกว่าวินโดว์ ได้ 8 ช่อง ที่ทับซ้อนกันดังรูปที่ 1 ในการทำงานจะมีกลไกการทำงานแบบกระบวนความโดยมีรีจิสเตอร์ 6 ตัวที่ทับซ้อนกัน การทับซ้อนของรีจิสเตอร์มีเพื่อให้การส่งพารามิเตอร์ผ่านทางรีจิสเตอร์เหล่านี้ได้ โดยสามารถส่งจากกระบวนความ A กับ B หรือ B กับ C โดยมีรีจิสเตอร์ที่ใช้ในแต่ละกระบวนความเป็นแบบโลคัล และใช้ r0-r9 เป็นโกลบอลรีจิสเตอร์ หรือกล่าวได้ว่า ภายในซีพียูมีกลไกการทำงานเสมือนมีสามโปรเซสเซอร์ที่ทำงานตามกระบวนความทั้งสามแบบขนานกันและส่งผ่านพารามิเตอร์ระหว่างกันผ่านทางรีจิสเตอร์ที่ทับซ้อนกัน

จากรายงานของ RISC I และ RISC II กล่าวว่า ผลของการทำงานจะ



รูปที่ 1 โครงสร้างรีจิสเตอร์ของ RISC I

มีความเร็วที่เพิ่มขึ้นอีก 2 ถึง 4 เท่า เมื่อเทียบกับการทำงานแบบ CISC เช่น VAX หรือ 68000

ทำไม RISC จึงทำงานได้เร็วขึ้น

ไม่นานนักจากที่ RISC I ได้รับการพัฒนาขึ้น ที่มหาวิทยาลัยสแตนฟอร์ดก็มีการพัฒนาซีพียูที่ใช้สถาปัตยกรรมแบบ RISC เช่นกัน แต่ให้ชื่อว่า MIPS ซึ่งเป็นตัวย่อมาจาก Microprocessor without Interlocked Pipe Stages ซึ่งเป็นซีพียูที่เน้นในเรื่องการทำงานที่รวดเร็วเช่นกัน

เพื่อให้การทำงานได้ตามวัตถุประสงค์ โครงสร้างการทำงานภายในจึงต้องใช้หลักการแบบขนาน ซีพียู RISC I มีลักษณะการวางรีจิสเตอร์สนับสนุนขบวนการทำงานภายใน 3 กระบวนความ โครงสร้างของ RISC ทั่วไปก็ต้องทำเช่นว่านี้โดยให้การทำงานมีลักษณะตามปรัชญาของ RISC ดังนี้

ทำงานได้ในไซเกิลเดียวต่อคำสั่ง

ความจริงแล้วการทำงานหนึ่งคำสั่งต้องใช้เวลามากกว่าหนึ่งไซเกิล แต่ที่ทำได้เพราะให้คำสั่งเหล่านั้นมีลักษณะเป็นไปป์และขนานกัน จึงทำให้เวลาโดยรวมเมื่อเฉลี่ยออกมาแล้วได้คำสั่งละไซเกิลเดียว

ออกแบบโหลด/สตอร์

เพื่อลดการทำงานที่ล่าช้า คำสั่งเหล่านี้จะต้องแน่นอนและทำงานได้ในลักษณะขนานตามขบวนการที่กำหนด

ใช้การควบคุมด้วยฮาร์ดแวร์

โดยอาศัยวงจรอิเล็กทรอนิกส์ที่ทำให้การทำงานเป็นไปในลักษณะที่ทำได้ในไซเกิลเดียว โดยไม่ต้องใช้ไมโครโค้ด

มีจำนวนคำสั่งน้อย และการอ้างแอดเดรสจำกัด

เพื่อให้โครงสร้างตายตัว ในเรื่องไซเกิลการทำงานจึงต้องลดจำนวนคำสั่งลงเหลือเท่าที่จำเป็น เพื่อให้ฮาร์ดแวร์ควบคุมได้

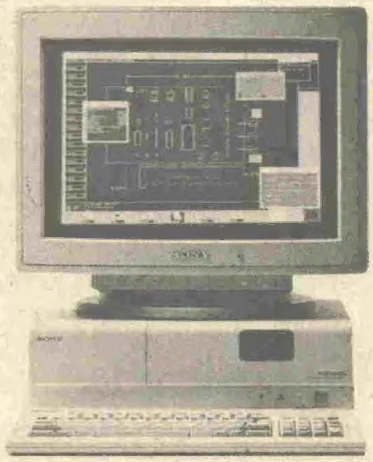
ฟอร์แมตคำสั่งง่าย

เพื่อการแปลความหมายและกระทำความด้วยฮาร์ดแวร์ได้ รูปแบบคำสั่งจึงต้องง่าย

ต้องใช้เวลาคอมพายล์ภาษาระดับสูงมาก

เพื่อเปลี่ยนแปลงภาษาระดับสูงมาเป็นภาษาระดับต่ำ ต้องใช้คอมพายเลอร์ที่แปลงมาเป็นรหัสภาษาเครื่องได้ โดยที่คำสั่งห่างไกลทางความหมายจากภาษาระดับสูงมาก ดังนั้นจึงต้องเสียเวลาในการคอมพายล์นานกว่า

การที่ RISC ทำงานได้เร็วกว่า เพราะการวางโครงสร้างการทำงานได้กำหนดส่วนของการทำงานไว้ชัดเจนและเป็นแบบขนาน เช่น MIPS โมเดล R2000 ได้วางโครงสร้างของคำสั่งแต่ละคำสั่งมีการทำงาน 5 ขั้นตอน ดังนั้นจึงวางโครงสร้างการทำงานแบบขนานไว้ 5 ระดับ โดยแบ่งแยกส่วนของการทำงานแต่ละคำสั่งเป็น



IF - การเฟตช์คำสั่ง (Instruction Fetch) คือ จังหวะที่เฟตช์คำสั่งจากหน่วยความจำมาตีความหมาย

RD - อ่านโอเปอร์แรนด์จากรีจิสเตอร์ในซีพียู

ALU - คือ การคำนวณในหน่วย ALU ตามรหัสของคำสั่งนั้น ๆ

MEM - การติดต่อกับหน่วยความจำ

WB - การเขียนข้อมูลผลลัพธ์ลงในรีจิสเตอร์

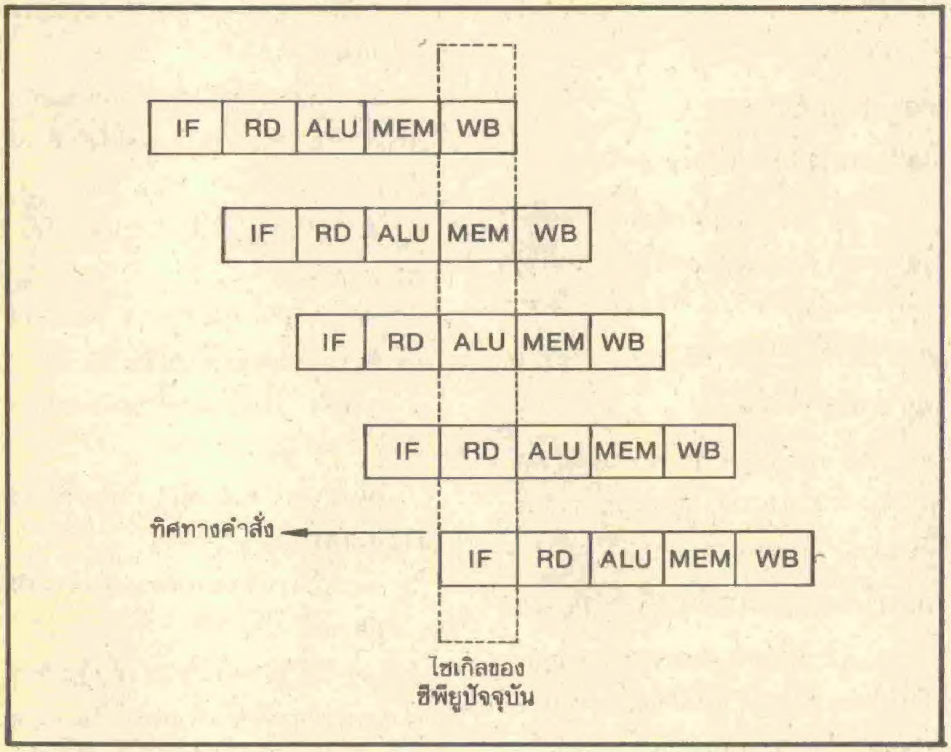
เมื่อนำมาเขียนแสดงการทำงานจะได้ดังรูปที่ 2

การใช้วิธีการของคำสั่งในการทำงานแบบขนานนี้ เป็นผลทำให้ความเร็วของการทำงานทำได้เสร็จในหนึ่งไซเกิล ทั้ง ๆ ที่แท้จริงแล้วต้องใช้ถึง 5 ไซเกิล แต่ทุกขณะจะมีส่วนของซีพียูทำงานขนานกันทั้ง 5 ส่วนนี้อยู่ โดยขณะที่เฟตช์คำสั่งหนึ่งอยู่ภายในก็มีการทำ RD, ALU, MEM และ WB ของแต่ละคำสั่งที่ผ่านมา ดังนั้นการเฟตช์จึงเกิดได้ทุกไซเกิล

กลไกการทำงานในลักษณะนี้ใช้กับซีพียู RISC ทุกตัว ตั้งแต่จุดเริ่มต้นของการพัฒนาจนถึงปัจจุบัน โครงสร้างของคำสั่งจึงอยู่ที่ ALU ว่าจะรับข้อมูลได้กว้างเพียงไร เช่น ถ้ารับข้อมูลได้ 64 บิต และให้ ALU เป็นหน่วยคำนวณทางคณิตศาสตร์แบบตัวเลขโฟลตติ้งพอยต์ก็ยิ่งทำได้เร็วยิ่งขึ้น เช่น ซีพียู i860 ของอินเทลก็ใช้หลักการของ RISC เช่นกัน

ชุดคำสั่งของ RISC

เพื่อให้เห็นโครงสร้างชุดคำสั่งที่



รูปที่ 2 แสดงการใช้ไฟล์ของคำสั่ง 5 ระดับใน R2000

ใช้ใน RISC ในที่นี้จะขอยกตัวอย่างชุดคำสั่งของ MIPS ไว้กระทำกับรีจิสเตอร์ 3 ตัว คือ scr1, scr2 และ dest กล่าวคือ scr เป็นรีจิสเตอร์ตัวทาง dest เป็นรีจิสเตอร์ผลลัพธ์ คำสั่งส่วนใหญ่เป็นคำสั่งพื้นฐานที่เข้าใจได้ง่าย

การออกแบบคำสั่งเหล่านี้มุ่งไปที่การใช้รีจิสเตอร์ภายใน ดังนั้นรีจิสเตอร์ภายในมักมีขนาดกว้าง ในที่นี้ใช้ขนาด 32 บิต (อินเทล i860 ใช้ขนาด 64 บิต) หากจะพิจารณาคำสั่งที่แสดงจะพบว่าคำสั่งเพียงเท่านี้ก็เพียงพอต่อการใช้งานหรือการเขียนโปรแกรมให้ทำงานในสิ่งต่าง ๆ ตามที่ต้องการได้แล้ว การออกแบบ RISC จึงเป็นสิ่งที่ใช้สถาปัตยกรรมที่แตกต่างจาก CISC โดยสิ้นเชิง

ความสามารถอยู่ที่การจัดการหน่วยความจำ

เมื่อซีพียู RISC ทำงานด้วยคำสั่งที่ใช้กับรีจิสเตอร์เป็นหลักมีเพียง LD กับ ST เท่านั้นที่ใช้จากหน่วยความจำ LD กับ ST จึงต้องเกี่ยวกับหน่วยความจำที่ซีพียูต้องติดต่อด้วยอย่างรวดเร็ว การที่ซีพียูต้องติดต่ออย่างรวดเร็วนี้จึงต้องอาศัยหน่วยความจำแคช ดังนั้นประสิทธิภาพของ RISC จึงขึ้นอยู่กับการจัดโครงสร้างของหน่วยความจำด้วย โดยที่หน่วยความจำแคชจะต้องมีบทบาทที่ทำให้ซีพียูติดต่อแล้วพบข้อมูลเป็นส่วนใหญ่

โครงสร้างของแคชที่ใช้กับ RISC เป็นแบบ direct mapped cache ใช้การติดต่อกับหน่วยความจำนี้ จะทำให้หน่วยความจำต่อกับแคชในลักษณะ

ตารางที่ 1 ชุดคำสั่งของ MIPS

คำสั่ง	โอเปอร์แรนด์	ความหมาย
Add	src1,src2,dest	Integer addition
And	src1,src2,dest	Logical AND
Ic	src1,src2,dest	Insert byte
Or	src1,src2,dest	Logical OR
Rlc	src1,src2;src3,dest	Rotate combined
Rol	src1,src2,dest	Rotate
Sll	src1,src2,dest	Shift left logical
Sra	src1,src2,dest	Shift right arithmetic
Srl	src1,src2,dest	Shift right logical
Sub	src1,src2,dest	Integer subtraction
Subr	src1,src2,dest	Reverse integer subtraction
Xc	src1,src2,dest	Extract byte
Xor	src1,src2,dest	Logical XOR
Ld	A[src],dst	Load based
Ld	[src1+src2],dst	Load based-indexed
Ld	[src1>>src2],dst	Load based-shifted
Ld	A,dst	Load direct

คำสั่ง	โอเปอร์แรนด์	ความหมาย
Ld	I,dst	Load immediate
Mov	stc,dst	Load immediate
St	src1,A[src]	Store based
St	src1,[src2+src3]	Store base-indexed
St	src1,[src2>>src3]	Store based-shifted
St	src,A	Store direct
Bra	dst	Unconditional relative jump
Bra	Cond,src1,src2,dst	Conditional jump
Jmp	dst	Unconditional jump direct
Jmp	A[src]	Unconditional jump based
Jmp	(A[src])	Unconditional jump indirect
Trap	Cond,src1,src2	Trap instruction
SavePC	A	Save multi-stage PC after trap or interrupt
Set	Cond,src,dst	Set conditional

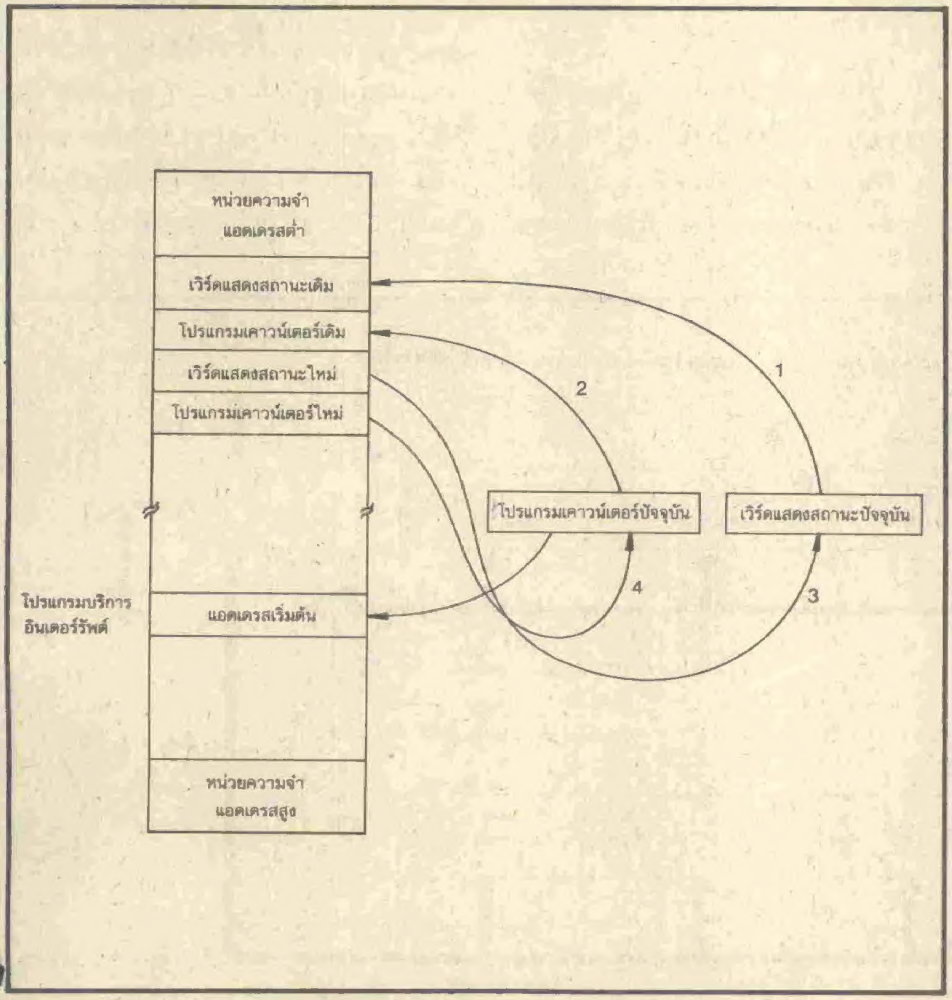
ที่มีการกำหนดตำแหน่งแน่นอน แคชแบบนี้จะทำให้การเข้าถึงทำได้เร็วและต้องการพื้นที่น้อยกว่า

โครงสร้างการอินเตอร์รัพต์ยังคงใช้เหมือนเดิม

ในซีพียูแบบ CISC ต้องอาศัยการทำงานในลักษณะโปรแกรมหลายระดับ โดยการใช้สัญญาณอินเตอร์รัพต์เข้ามาเป็นตัวสวิตช์ การทำงานในแต่ละส่วนของโปรแกรม

ในกรณีของ RISC ก็เช่นกัน การอินเตอร์รัพต์ก็มีกลไกการทำงานคล้ายกัน จะแตกต่างกันเพียงเล็กน้อยในเรื่องการทำงานภายในซีพียู ในเรื่องที่ CISC อาจมีไมโครโค้ดคอยจัดการบางอย่างให้ แต่แนวความคิดของการใช้งานก็ยังคงเหมือนกันดังรูปที่ 3

เมื่ออินเตอร์รัพต์เกิดขึ้น จะมีลำดับการทำงานโดยที่ซีพียูจะตรวจสอบสถานะของตนเองว่าอยู่ในลำดับที่จะตอบสนองต่ออินเตอร์รัพต์ได้หรือไม่ ถ้าพร้อมก็จะเริ่มทำงานโดยที่นำสถานะของโปรแกรมที่อยู่ใน PSW ไปเก็บในหน่วยความจำ แล้วนำค่าของโปรแกรมเคาวน์เตอร์ไปเก็บในโปรแกรมด้วย จากนั้นจึงรับค่าของเวิร์ดแสดงสถานะและโปรแกรมเคาวน์เตอร์ตัวใหม่ เพื่อเริ่มต้นทำงานในส่วนของโปรแกรมบริการต่อไป



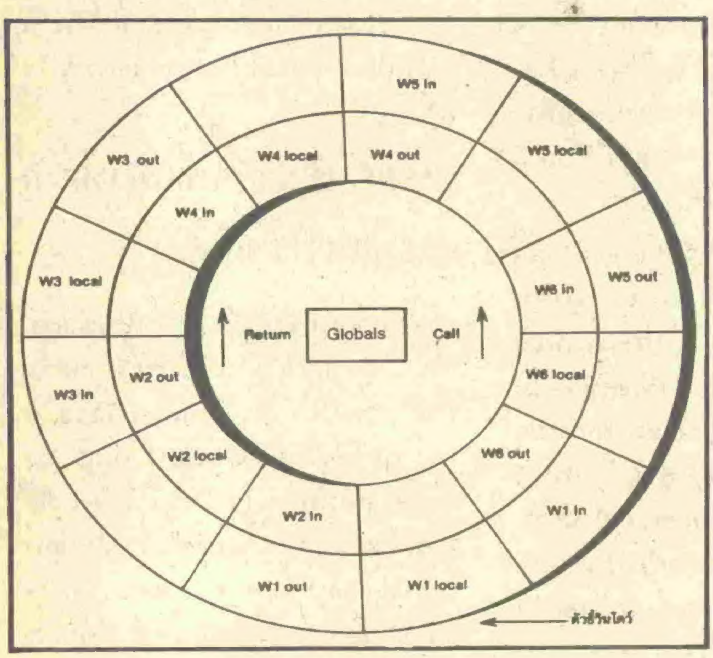
รูปที่ 3 แสดงกลไกการตอบสนองต่ออินเตอร์รัพต์

SPARC เป็นซีพียูที่แพร่หลายและมีแนวโน้มที่ดี

หลังจากที่บริษัท ซันไมโครซิสเต็ม ประสบความสำเร็จในเรื่องเวอร์กสเตชันที่ใช้ UNIX เป็นโปรแกรมจัดระบบงาน ซันใช้ซีพียูแบบ CISC ในตระกูล 68000 ก่อนพัฒนาการของ 68000 ได้ก้าวเข้ามาจนถึง 68030 ซันจึงได้พัฒนาและนำเอาความคิดของ RISC มาใช้ คือ SPARC สปาร์กเป็นซีพียูที่ทำงานได้เร็ว ปัจจุบันพัฒนาการของสปาร์กทำให้ซีพียูของซันทำงานได้เร็วกว่า 30 mips โครงสร้างของสปาร์กมีการจัดการรีจิสเตอร์ภายในออกเป็น 8 กลุ่ม กลุ่มละ 32 รีจิสเตอร์ โดยแบ่งเป็นรีจิสเตอร์ร่วมโกลบอล 8 ตัว และใช้ทำงาน 24 ตัว โครงสร้างการแบ่งออกมาเป็น 6 โปรเซสเพื่อแยกการทำงานแบบขนาน โครงสร้างของรีจิสเตอร์ทำงาน 24 ตัวมีการทับซ้อนกัน โดยแบ่งเป็นกลุ่ม ๆ ละ 8 รีจิสเตอร์ เรียกว่า In รีจิสเตอร์ โลคัลรีจิสเตอร์ และ Out รีจิสเตอร์ การทำงานจึงแบ่งเป็นวินโดว์ มีตัวชี้วินโดว์ที่กำลังดำเนินการอยู่ โครงสร้างของรีจิสเตอร์แสดงดังรูปที่ 4

พัฒนาการด้วยเทคโนโลยี VLSI

ความสำเร็จของ RISC ที่เห็นอยู่ทุกวันนี้ไม่ว่าจะเป็นสปาร์ก MIPS หรือ RISC/6000 ของไอบีเอ็ม ล้วนแล้วแต่เป็นผลมาจากเทคโนโลยีของชิพ



รูปที่ 4 โครงสร้างการจัดแบ่งรีจิสเตอร์ของสปาร์ก

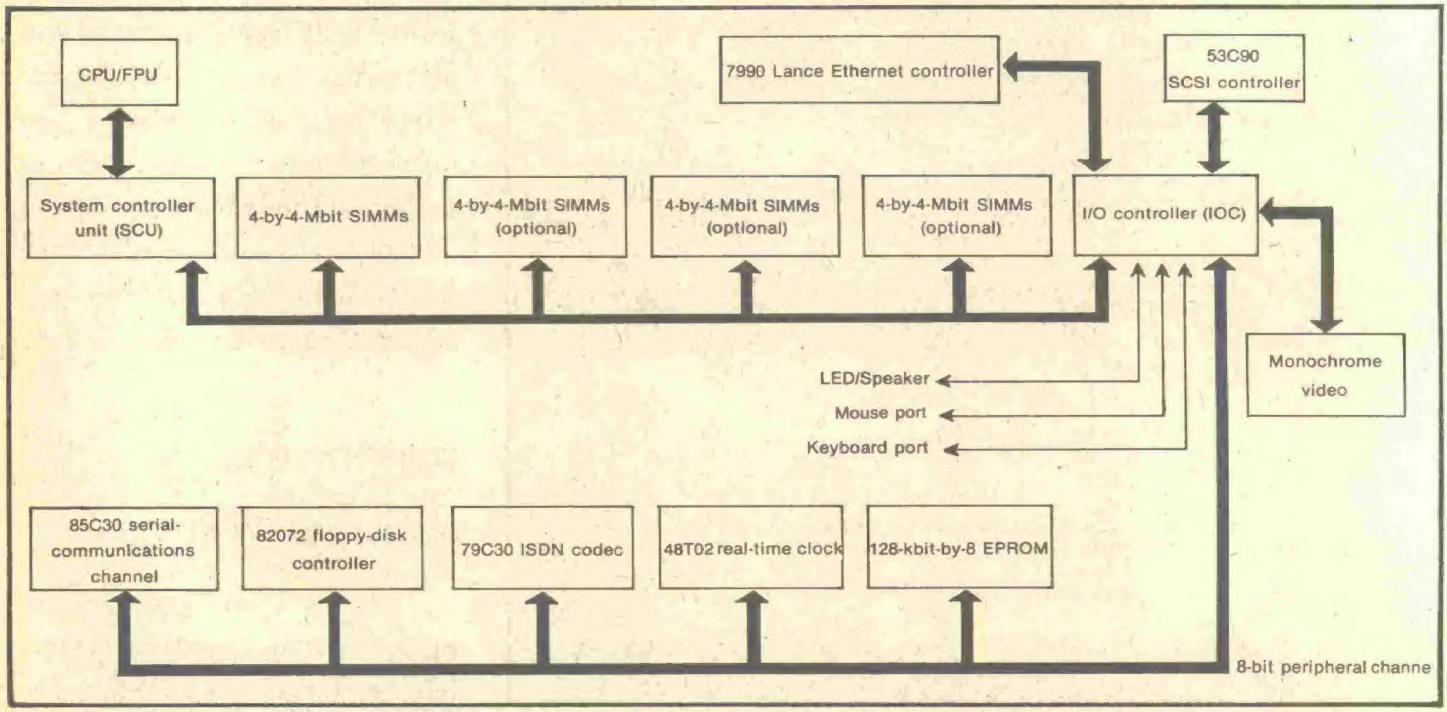
ที่ทำให้ชิพมีความหนาแน่นของตัวทรานซิสเตอร์ได้สูง ซีพียู RISC จึงมีแนวโน้มทางด้านราคาลดลง และจะทำให้เวอร์กสเตชันมีราคาถูกลงเทียบเคียงกับพีซี

รูปที่ 5 เป็นไดอะแกรมของเครื่อง SPARC SLC ซึ่งเป็นเวอร์กสเตชันราคาถูก สังเกตว่าทั้งระบบที่มีราคาไม่อยู่ที่ซีพียูอย่างเดียว แต่จะขึ้นกับอุปกรณ์ประกอบอื่น ๆ อีกหลายอย่าง

กำลังจะกลายเป็นพีซีในที่สุด

ด้วยการที่ผลิตชิพจำนวนมาก ชิพหรือกลุ่มชิพที่จะนำมาใช้ในเครื่องกลุ่ม RISC จึงไม่แตกต่างจากพีซีที่ใช้ CISC เช่น 80486 อีกต่อไปแล้ว ชิพ SPARC ตัวหนึ่งอาจมีราคาถูกกว่าชิพของอินเทล 80486 โครงสร้างของคอมพิวเตอร์พีซีก็ไปใช้ชิพที่ซับซ้อนยิ่งขึ้น ในขณะที่เวอร์กสเตชันหันมาใช้ RISC

หนทางของ RISC ยังอีกไกล เพราะแม้แต่ไอบีเอ็มก็มีซีพียู RISC ของตัวเอง ขีดความสามารถทางความเร็วของ RISC จึงเป็นที่กล่าวถึงกันอย่างมาก และเชื่อว่า RISC จะมีขีดความสามารถอีกหลาย ๆ อย่างเพิ่มขึ้น ไม่ว่าจะเป็นเรื่องการคำนวณ เรื่องความเร็ว RISC จึงเป็นซีพียูที่เน้นความเหมาะสมทางด้านวิทยาศาสตร์มากกว่า



รูปที่ 5 บล็อกไดอะแกรมของเวอร์กสเตชัน SPARC SLC